CLAIMS

What is claimed is:

1. A $2^n \times 2^n$ generalized divide-and-conquer network, n>1, comprising

5 $2^{\lfloor n/2 \rfloor} \, 2^{\lceil n/2 \rceil} \times 2^{\lceil n/2 \rceil}$ input nodes,

$2^{\lceil n/2 \rceil} \, 2^{\lfloor n/2 \rfloor} \times 2^{\lfloor n/2 \rfloor}$ output nodes, and

an interstage exchange connecting the input nodes to the output nodes.

2. The generalized divide-and-conquer network as recited in claim 1

10   wherein the interstage exchange is a bit-permuting exchange induced by a permutation σ

on integers from 1 to n such that σ maps the numbers $\lfloor n/2 \rfloor+1, \lfloor n/2 \rfloor+2, \ldots, n$, into the set {1,

2, ..., $\lceil n/2 \rceil$} excluding the bit-permuting exchange equal to the $\lfloor n/2 \rfloor^{\text{th}}$ power of $\text{SHUF}^{(n)}$.

3. The generalized divide-and-conquer network as recited in claim 2

15   wherein

each of the $2^{\lfloor n/2 \rfloor} \, 2^{\lceil n/2 \rceil} \times 2^{\lceil n/2 \rceil}$ input nodes is a $2^{\lceil n/2 \rceil} \times 2^{\lceil n/2 \rceil}$ generalized

divide-and-conquer network, and

each of the $2^{\lceil n/2 \rceil} \, 2^{\lfloor n/2 \rfloor} \times 2^{\lfloor n/2 \rfloor}$ input nodes is a $2^{\lfloor n/2 \rfloor} \times 2^{\lfloor n/2 \rfloor}$ generalized

divide-and-conquer network.

4. The generalized divide-and-conquer network as recited in claim 2

wherein the bit-permuting exchange is a $SWAP^{(n, r)}$ exchange.

5

5. A $2^n \times 2^n$ generalized divide-and-conquer network associated with a n-leaf

balanced binary tree, n>1, the network comprising

$$2^{\lfloor n/2 \rfloor} 2^{\lceil n/2 \rceil} \times 2^{\lceil n/2 \rceil} \text{ input nodes,}$$

$$2^{\lceil n/2 \rceil} 2^{\lfloor n/2 \rfloor} \times 2^{\lfloor n/2 \rfloor} \text{ output nodes, and}$$

10　　　　an interstage exchange connecting the input nodes to the output nodes,

wherein the interstage exchange is a bit-permuting exchange induced by a permutation σ

on integers from 1 to n such that σ maps the numbers $\lfloor n/2 \rfloor +1, \lfloor n/2 \rfloor +2, ..., n$, into the set { 1,

2, ..., $\lceil n/2 \rceil$} excluding the bit-permuting exchange equal to the $\lfloor n/2 \rfloor^{th}$ power of $SHUF^{(n)}$.

15　　　　6. A $2^n \times 2^n$ generalized divide-and-conquer network, n>1, achieving an optimal

layout complexity under the 2-layer Manhattan model with reversed layers and optimal

structural modularity among all $2^n \times 2^n$ banyan-type networks, the network comprising

$$2^{\lfloor n/2 \rfloor} 2^{\lceil n/2 \rceil} \times 2^{\lceil n/2 \rceil} \text{ input nodes,}$$

$2^{\lceil n/2 \rceil} 2^{\lfloor n/2 \rfloor} \times 2^{\lfloor n/2 \rfloor}$ output nodes, and

an interstage exchange connecting the input nodes to the output nodes,

wherein the interstage exchange is a bit-permuting exchange induced by a permutation σ

on integers from 1 to n such that σ maps the numbers $\lfloor n/2 \rfloor + 1, \lfloor n/2 \rfloor + 2, \ldots, n$, into the set { 1,

5   2, ..., $\lceil n/2 \rceil$} excluding the bit-permuting exchange equal to the $\lfloor n/2 \rfloor^{th}$ power of $SHUF^{(n)}$.

7. A method for recursively constructing a $2^n \times 2^n$ generalized divide-and-conquer network, n>1, comprising

forming the bit-permuting 2-stage tensor product between a generalized

10   $2^{\lceil n/2 \rceil} \times 2^{\lceil n/2 \rceil}$ divide-and-conquer network and a generalized $2^{\lfloor n/2 \rfloor} \times 2^{\lfloor n/2 \rfloor}$ divide-and-conquer

network, and

recursively, each $2^k \times 2^k$ generalized divide-and-conquer network (k<n) is

constructed by forming the bit-permuting 2-stage tensor product between a generalized

$2^{\lceil k/2 \rceil} \times 2^{\lceil k/2 \rceil}$ divide-and-conquer network and a generalized $2^{\lfloor k/2 \rfloor} \times 2^{\lfloor k/2 \rfloor}$ divide-and-conquer

15   network, until k=1, where a 2×2 generalized divide-and-conquer network is a single cell.

8. The method as recited in claim 7 wherein the forming includes

configuring a first stage of input nodes where each of the input nodes is a

- 228 -

generalized $2^{\lceil k/2 \rceil} \times 2^{\lceil k/2 \rceil}$ divide-and-conquer network,

configuring a second stage of output nodes where each of the output nodes

is a generalized $2^{\lfloor k/2 \rfloor} \times 2^{\lfloor k/2 \rfloor}$ divide-and-conquer network, and

interconnecting the first stage and the second stage by a bit-permuting

5   exchange induced by a permutation σ on integers from 1 to k such that σ maps the numbers

$\lfloor k/2 \rfloor + 1, \lfloor k/2 \rfloor + 2, \ldots, k$, into the set $\{1, 2, \ldots, \lceil k/2 \rceil\}$ excluding the bit-permuting exchange

equal to the $\lfloor k/2 \rfloor^{\text{th}}$ power of $\text{SHUF}^{(n)}$.

9. A method for recursively constructing a $2^n \times 2^n$ generalized divide-and-conquer

10   network, n>1, in correspondence to an n-leaf balanced binary tree, the method comprising

constructing, in correspondence to the root R of the tree, the generalized

$2^n \times 2^n$ generalized divide-and-conquer network by forming the bit-permuting 2-stage

tensor product between a generalized $2^p \times 2^p$ divide-and-conquer network which is

associated with the left-son of R having a weight of p and a generalized $2^q \times 2^q$

15   divide-and-conquer network which is associated with the right-son of R having a weight of

q, with $|p-q| \leq 1$ and wherein $p = \lceil n/2 \rceil$ and $q = \lfloor n/2 \rfloor$, or $p = \lfloor n/2 \rfloor$ and $q = \lceil n/2 \rceil$, and

recursively, in correspondence to a generic internal node H with weight k

until k=1 and wherein a 2×2 generalized divide-and-conquer network is a single cell.,

constructing a $2^k \times 2^k$ generalized divide-and-conquer network (k<n) by forming the

bit-permuting 2-stage tensor product between a generalized $2^s \times 2^s$ divide-and-conquer

network which is associated with the left-son of H having a weight of s and a generalized

$2^t \times 2^t$ divide-and-conquer network which is associated with the right-son of H having a

5   weight of t, with |s−t|≤1 and wherein s=⌈k/2⌉ and t=⌊k/2⌋, or s=⌊k/2⌋ and t=⌈k/2⌉.